REMARKS

Upon entry of the following Response, claims 1-22 remain pending in the present application. Applicants respectfully request reconsideration of claims 1-22 in view of the following remarks.

In item 3 of the Office Action, claims 1-4, 9-12, and 17-20 have been rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent 6,367,068 issued to Vaidyanathan et al. (hereafter "Vaidyanathan"). It is axiomatic that anticipation under Section 102 "requires the disclosure in a single prior art reference of each element of the claim under construction" W.L. Gore & Associates, Inc. v. Garlock, Inc., 220 USPQ 303, 313 (Fed. Cir. 1983). For the reasons that follow, Applicants respectfully assert that Vaidyanathan fails to show or suggest all of the elements of claims 1-4, 9-12 and 17-20. Accordingly, Applicants request that the rejection of claims 1-4, 9-12 and 17-20 be withdrawn.

To begin, reference is made to claim 1 that states:

> 1. A method for parsing a markup file, comprising:
> parsing a first portion of the markup file with a lightweight parser in a computer system, the lightweight parser being capable of performing a first set of parsing tasks;
> parsing a second portion of the markup file with a heavyweight parser in the computer system, the heavyweight parser being capable of performing a second set of parsing tasks, wherein the first set of parsing tasks is a subset of the second set of parsing tasks; and
> transitioning between the parsing of the first portion of the markup file with the lightweight parser to the parsing of the second portion of the markup file with the heavyweight parser upon an occurrence of a transition event.

With respect to claims 1, 9 and 17, the Office Action states:

> "Vaidyanathan discloses a method for parsing a markup file, comprising: parsing a first portion of the markup file with a lightweight parser in a computer system, the lightweight parser being capable of performing a set of parsing tasks (col 6, lines 20-35); parsing a second portion of the markup file with a heavyweight parser in the computer system (col 8, lines 20-35), the heavyweight parser being capable of performing a second set of parsing tasks, wherein the first set of parsing tasks is a subset of the second set of parsing tasks (col 8, lines 20-45; col 11, lines 30-40); and transitioning between the parsing of the first portion of the markup file with the lightweight parser to the parsing of the second portion of the markup file with the heavyweight parser upon an occurrence of a transition event (col 9, lines 25-50)." (Office Action, page 3).

Applicant respectfully disagrees. Specifically, Vaidyanathan does not show or suggest the concept of employing heavyweight and lightweight parsers as claimed in claim 1. The parser of Vaidyanathan parses source code while it is being written or edited by a user. Specifically, Vaidyanathan states:

> "As the source code is written (developed) by a computer programmer via the editor, the dynamic parser continuously (dynamically) checks the entered source code for errors. Unlike the prior art, the invention does not have to wait for the programmer to initiate building or compiling him or herself." (Col. 6, lines 15-20)

The parsing of Vaidyanathan is performed by a **single parser** that is used to parse source code in two different stages. In each stage, different portions of the source code are parsed by the same parser (204 or 402). The parsing is done during idle cycles when a programmer pauses during the writing of the source code. (See col. 5, lines 56-67). In this respect, Vaidyanathan fails to show or suggest a lightweight parser that performs a first set of parsing tasks and a heavyweight parser that performs a second set of parsing tasks, where the first set of parsing tasks is a subset of the second set of parsing tasks.

Also, since Vaidyanathan only discusses the use of a single parser, there is no transition between the parsing of the first portion of the markup file with the lightweight parser to the parsing of the second portion of the markup file with the heavyweight parser upon an occurrence of a transition event as claimed in claim 1. Specifically, at column 9, lines 25-50, Vaidyanathan states:

> "When the (secondary) parser thread encounters an entry in the first stage queue it obtains the current contents of the source code file from the editor rather than the one stored on the hard disk drive. Since the parser (secondary thread) and the editor are in two different threads, and the editor line tables may not be thread-safe, as known within the art, the parser may not be able to directly access the editor's line table. Therefore, it posts a message to the primary (UI/editor) thread, to copy the contents of the current editor's line table into a shared memory location. After it posts this message, it waits for an event (a signal) from the editor before proceeding further. The editor, on receipt of this message, copies the current contents to the specified memory location, resets the aforementioned dirty bit, and then fires the event. The parser thread, which was blocked on this particular event, wakes up, and hands over a pointer to the shared memory location to the compiler for parsing.

> Once the first stage of the parsing is complete, the NCB database/file is updated, as has been described, and the parser thread generates a notification to so indicate. The second stage of parsing then begins as described in this application's to identify errors within the source code. Such errors are passed along to the editor, so that the editor may be able to identify them to the programmer on-screen.

Applicant asserts that the above excerpt addresses the interplay between an editor application and a parser. Specifically, the interplay between the editor and the parser is described so that the parser can perform parsing functions while a programmer generates the source code. Nowhere is there a transition from a lightweight parser to a heavyweight parser based upon any event.

Therefore, Applicant asserts that Vaidyanathan fails to show or suggest all of the elements of claim 1 and all of the elements of claims 9 and 17 as including subject matter that is similar in scope with the subject matter of claim 1. Accordingly, Applicant requests that the rejection of claims 1, 9, and 17 be withdrawn. In addition, Applicant requests that the rejection of claims 2-4, 10-12, and 18-20 be withdrawn as depending from claims 1, 9, and 17, respectively.

In addition, claim 2 includes the element of "detecting an occurrence of the transition event comprising a requirement that the lightweight parser perform a parsing task excluded from the first set of parsing tasks." In addition, claims 10 and 18 include elements similar in scope with that of claim 2 described herein. With respect to claims 2, 10, and 18, the Office Action states "Vaidyanathan discloses the limitations as discussed in claims 1, 9, and 17 above. Vaidyanathan further discloses detecting an occurrence of a transition event comprising a requirement that the lightweight parser perform a parsing task excluded from the first set of parsing tasks (col. 9, lines 50-60)." (Office Action, page 4).

Applicant respectfully disagrees. Specifically, at column 9, lines 50-60, Vaidyanathan states:

> The second stage action described here is error diagnosis. Other actions are described in copending, cofiled, and coassigned applications previously incorporated by reference. One possible second stage action that can be embodied is doing a dynamically traditional compile at idle time. This has not been described in detail in these applications.

Applicant asserts that the above excerpt has nothing to do with the concept of detecting an occurrence of a transition event that comprises a requirement that the

lightweight parser performs a parsing task excluded from the first set of parsing tasks. In addition, given that there is only a single parser described in Vaidyanathan to begin with, how is it that the parser may be required to perform parsing tasks that are excluded from a set of tasks for which it is designed, thereby causing a transition to a second parser?

Consequently, Applicant asserts that the rejection of claims 2, 10, and 18 is improper. Accordingly, Applicant requests that the rejection of claims 2, 10, and 18 be withdrawn for these additional reasons.

CONCLUSION

Applicants respectfully request that all outstanding objections and rejections be withdrawn and that this application and all presently pending claims be allowed to issue. If the Examiner has any questions or comments regarding Applicants' response, the Examiner is encouraged to telephone Applicants' undersigned counsel.

Respectfully submitted,

Michael J. D'Aurelio
Reg. No. 40,977

D'Aurelio & Mathews, LLC
96 Church Street
Chagrin Falls, Ohio 44022
Phone: (440) 729-7450
Fax: (440) 729-7465